LA-UR-03-6275

*Approved for public release;
distribution is unlimited.*

Title:

Accelerator Validation of an FPGA SEU Simulator

Author(s):

D. Eric Johnson, LANL, BYU
Michael Caffrey, LANL
Paul Graham, LANL
Nathan Rollins, BYU
Michael Wirthlin, BYU

Submitted to:

IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 50, NO. 6, DECEMBER 2003

Accelerator Validation of an FPGA SEU Simulator

Eric Johnson, Michael Caffrey, Paul Graham, Nathan Rollins, Michael Wirthlin

***Abstract*— An accelerator test was used to validate the performance of an FPGA Single Event Upset (SEU) simulator. The Crocker Nuclear Laboratory cyclotron proton accelerator was used to irradiate the SLAAC1-V, a Xilinx Virtex FPGA board. We also used the SLAAC1-V as the platform for a configuration bitstream SEU simulator. The simulator was used to probe the "sensitive bits" in various logic designs. The objective of the accelerator experiment was to characterize the simulator's ability to predict the behavior of a test design in the proton beam during a dynamic test. The test utilized protons at 63.3 MeV, well above the saturation cross-section for the Virtex part. Protons were chosen because, due to their lower interaction rate, we can achieve the desired upset rate of about one configuration bitstream upset per second. The design output errors and configuration upsets were recorded during the experiment and compared to results from the simulator. In summary, for an extensively tested design, the simulator predicted 97% of the output errors observed during radiation testing. The SEU simulator can now be used with confidence to quickly and affordably examine logic designs to 'map' sensitive bits, to provide assurance that incorporated mitigation techniques perform correctly, and to evaluate the costs and benefits of various mitigation strategies. The simulator provides an excellent test environment that accurately represents radiation induced configuration bitstream upsets.**

***Index Terms*— SEU, FPGA, simulator, proton accelerator, radiation, dynamic testing**

I. Introduction

THERE is increasing interest in the use of SRAM-based Field Programmable Gate Arrays (FPGAs) in space-based applications such as remote sensing[1]. FPGAs are programmable logic devices, which allow the user to specify the function to be performed. There are many available resources within an FPGA to perform various logic functions. The way in which these resources are utilized and interconnected is specified by the circuit design, also known as a configuration bitstream. The configuration bitstream determines which resources within the FPGA are used to implement a specific logic design. The configuration is sensitive to SEUs, some of which will result in changes to the design. This concept is illustrated in Figure 1. SRAM based FPGAs can be reprogrammed quickly (<<1 sec) and indefinitely with new configuration bitstreams. This feature is exploited extensively in this work for error introduction. It is also a compelling capability for deployed systems.

Reconfigurable FPGAs within a spacecraft allow the use of digital circuits that are both application-specific and reprogrammable. Unlike application-specific integrated circuits (ASICs), FPGAs can be configured *after* the spacecraft has been launched. This flexibility allows the same FPGA resources to be used for multiple instruments, missions, or changing spacecraft objectives. Errors in an FPGA design can be resolved by fixing the incorrect design and reconfiguring the FPGA with an updated configuration bitstream.

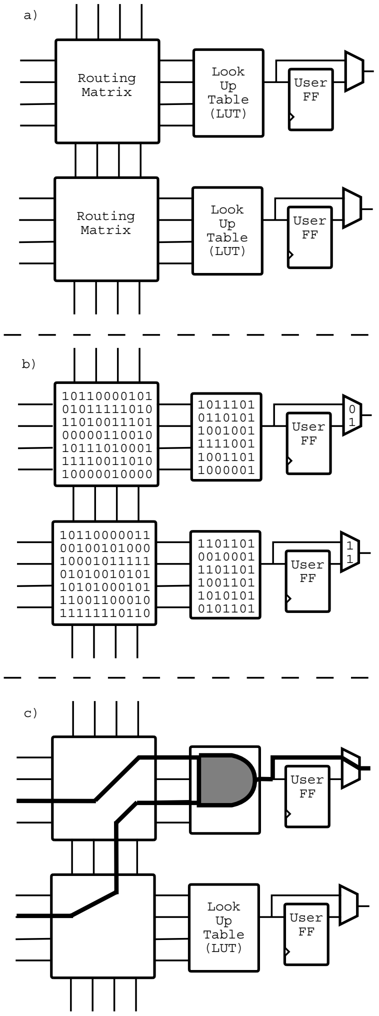


Fig. 1. a) Example FPGA architecture. b) The configuration bitstream defines device function. c) An example circuit

While the use of reprogrammable FPGAs for spaced-based applications offers a number of important advantages, these SRAM-based FPGAs are very sensitive to both heavy ion and proton induced single event upsets (SEUs)[2]. Such upsets affect all types of internal FPGA state including user design flip-flops, the FPGA configuration bitstream, and half-latches[3],[4]. Upsets in the FPGA configuration bitstream are especially problematic because they can change the actual

Manuscript received August 29, 2003. This work was supported by the Department of Energy at Los Alamos National Laboratory.

P. Graham and M. Caffrey are with Los Alamos National Laboratory, Los Alamos, NM.

D. E. Johnson is with both Los Alamos National Laboratory and Brigham Young University.

N. Rollins and M. Wirthlin are with the Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT.

circuit design. The configuration bitstream represents the bulk of the state in the device, so understanding the behavior of the device in the presence of SEUs is important.

A configuration bitstream SEU simulator was created to study the impact of upsets within the configuration memory on FPGA designs[5][6]. This simulator is built on the SLAAC1-V PCI FPGA board[7]. Radiation-induced upsets are simulated by artificially changing the contents of the configuration memory through device partial reconfiguration. The output of the device is subsequently monitored to determine the impact of a given configuration memory upset on design behavior. An important benefit of this SEU simulator is that it facilitates dynamic circuit testing without the need of expensive and cumbersome ground-based radiation tests. While this SEU simulator is convenient, it is essential to insure that the simulator results match those measured from tests with radiation sources. Dynamic testing was conducted at the Crocker Nuclear Laboratory on the SLAAC1-V FPGA board to validate the accuracy of the SEU simulator. During accelerator testing, we wanted to obtain approximately one configuration upset per observation cycle. A proton radiation source was chosen because protons have a low interaction rate, making this upset rate feasible.

This paper begins with a discussion of the configuration bitstream SEU simulator architecture, functionality, and performance. A description of the designs tested is also included, along with the steps necessary to prepare a design for simulator or radiation testing. The radiation testing procedure is then presented, followed by a discussion of results obtained with the simulator and through accelerator testing. Finally, an evaluation is made of the accuracy and usefulness of the configuration SEU simulator.

II. SEU Simulator

A configuration bitstream SEU simulator was created to test the behavior of FPGA designs in the presence of SEUs within the configuration memory[5]. Because all information about an FPGA design is stored in the configuration bitstream, whenever the state of a bit within the configuration memory is upset the function of the FPGA design may change. Signals may be rerouted, logic functions changed, or even the clock disconnected. The simulator monitors a design to detect if a configuration upset causes an output error.

The SEU simulator is based on the SLAAC1-V board, a high-speed FPGA board containing three Xilinx Virtex 1000 FPGAs. The architecture of the SLAAC1-V board is shown in Figure 2. PE0 is used to provide stimulus to designs in PE1 and PE2 which operate synchronously and, under normal circumstances, behave identically. During SEU simulation, the configuration memory of PE1, the design under test (DUT), is artificially upset through partial reconfiguration, which changes the contents of the configuration bitstream. The design is then executed to determine its true behavior in the presence of a configuration bitstream SEU. PE0 monitors the circuit outputs of PE1 and PE2 to determine if the introduction of an artificial configuration upset into the bitstream of PE1 has caused an output error. If so, the configuration bit which was upset is marked as a sensitive location. This entire process is outlined in Figure 3.

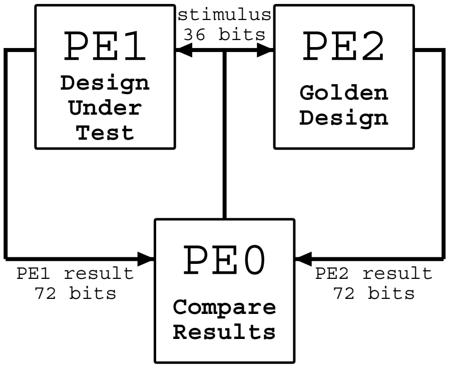


Fig. 2. SLAAC-1V Configuration SEU Simulator architecture.

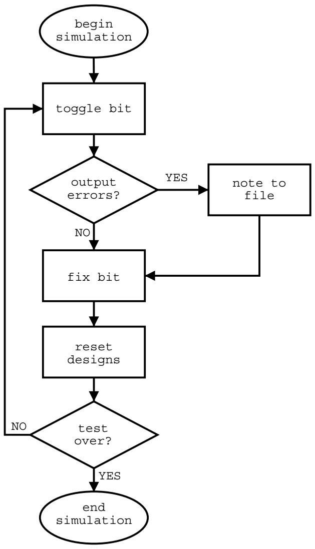


Fig. 3. Simulator Flow Diagram.

The distribution of the total static cross-section for the Virtex 1000 part is shown in Figure 4. There are $5.8x10^6$ configuration bits and $2.5x10^4$ flip flop bits with a saturation cross-section of $0.022x10^{-12}cm^2$ per bit for protons[2]. This yields a total device static cross-section for protons of $128x10^{-9}cm^2$. Clearly the configuration bitstream dominates the static sensitivity. This is important because the simulator tests only the configuration bitstream, it is not able to directly alter user flip-flop state. Radiation tests cause upsets within the entire static cross-section of the device, both within the configuration bitstream and user flip-flops. However, because the configuration bitstream dominates the static cross-section (more than 99%) , the SEU simulation results are relevant for characterizing the dynamic behavior of a design under the influence of SEUs. It is important in the comparison of the simulator and the accelerator to understand that the results are not identical, but approximate due to the small percentage (0.4%) of the cross-section that is not covered in simulation.

In the process of a typical configuration SEU simulation, each of these bits is individually upset to determine the effect

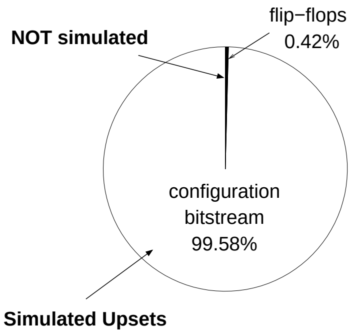


Fig. 4. Distribution of the static cross-section of the Virtex 1000. There are $5.8x10^6$ configuration bits tested by the simulator and the $2.5x10^4$ flip-flops which are untested in simulation.

they have upon the functionality of a given design. Because there is such a large number of configuration bits within the bitstream, it is essential that this test be performed as quickly as possible. The current testing procedure with the SLAAC1-V board requires 215 μs to test a single configuration bit, meaning that the entire configuration bitstream can be tested in approximately 20 minutes.

A. Designs Tested and their Preparation

Two types of designs have been tested with the SEU simulator. The first, a pipelined multiply-and-add design, emphasizes datapath characteristics, as it is a feed-forward design. The second, a linear feedback shift register design (LFSR) was created to study the effects of SEUs on a design with feedback.

The pipelined multiply-and-add design, as illustrated in Figure 5, consists of eight separate multipliers, whose outputs are summed to give a final result. The 72-bit wide version of this design has 36-bit wide A and B inputs, with an output which is 72-bits wide. The problem of having only 36 bits available for design input (see Figure 2) was resolved by having input A receive all 36 bits in their original bit ordering, with input B receiving a permuted version of those same 36 bits. A 36-bit wide version of this same design was created in order to study the effects of device utilization upon SEU sensitivity for dynamically tested designs.

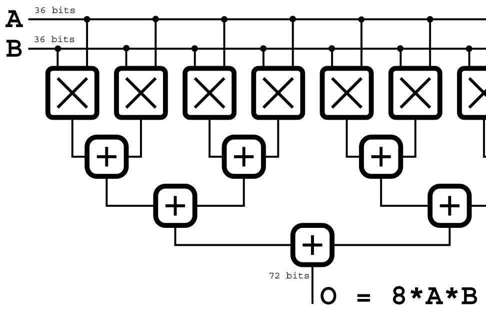


Fig. 5. 72Mult test design.

The LFSR design consists of clusters of 20-bit wide LFSRs, with taps at locations given by [8]. Each LFSR cluster contains six LFSRs, whose outputs are XORed together to form one bit of the final circuit output. A 72-bit wide LFSR design was created, meaning that this design held 72 LFSR clusters, one for each bit of the output. This design is illustrated in Figure 6.

Design	Slices	LUTs	Flip-Flops
72Mult	8,308	10,872	15,264
36Mult	2,206	2,844	3,744
72LFSR	8,712	576	8,640
XCV1000	12,288	24,576	24,576

TABLE I

DEVICE UTILIZATION FOR THREE TEST DESIGNS. THE LAST ROW SHOWS THE RESOURCES AVAILABLE IN THE XCV1000 FPGA.

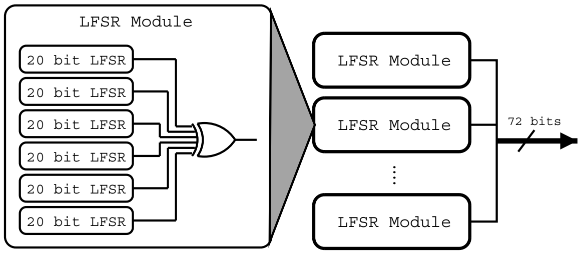


Fig. 6. 72-bit Linear Feedback Shift Register design.

The device utilization for each of the three test designs created is given in Table I. The fourth row shows the resources available in the XCV1000 FPGA. We chose test designs with different utilization rates and logic types to illustrate that the dynamic sensitivity of the part is a subset of the static cross-section. We anticipated that smaller designs would be less sensitive, and that designs with different architectures, such as the multiplier and LFSR, would have different sensitivities. The multiplier test designs are LUT intensive compared to the LFSR. The 36Mult is approximately 26% the size of the 72Mult test design, but with the same logic architecture.

Proper steps must be taken to prepare a design to be successfully tested with the configuration bitstream SEU simulator. All half-latches must be effectively disconnected from the design. This is done with the use of a half-latch removal tool[3]. Also, because readback is used to monitor the state of the configuration bitstream during radiation testing, no LUT RAMs may be present in the design[9]. If these two steps are not taken, inconsistent results may be obtained.

B. Simulator Example

For every design tested, the simulator generates a database containing the probability that an SEU at a given bitstream location will cause an output error. The probability of failure for a bitstream location is computed by dividing the total number of output errors observed for that location by the number of artificial upsets that were inserted. The sequence in which artificial configuration upsets are injected into the bitstream has no observable impact on the sensitivity of a given bit. All designs subjected to SEU simulation were operated at 20 MHz.

A look at the simulator results conducted on the 72Mult test design shows that a definite relationship exists between device utilization and design sensitivity. Figure 7, a circuit schematic of the 72Mult design, illustrates the device utilization of the FPGA. For comparison, a map of sensitive locations within the design can be created by plotting the row and column of the bit offset of sensitive bits. Figures 8 and 9 are graphical views of this ‘sensitivity map’ generated from the database for the 72Mult design. Figure 10 is a probability distribution function of those bits that were found to be sensitive.

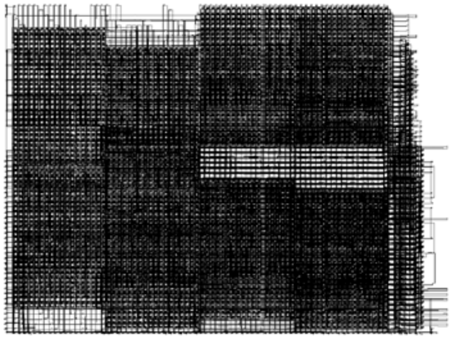


Fig. 7. Routing within the 72Mult test design taken from FPGA Editor.

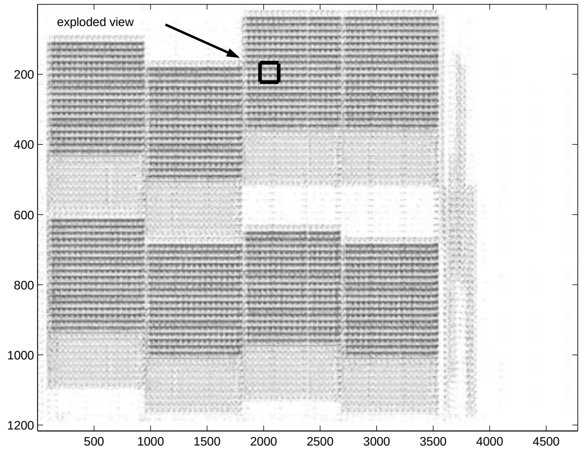


Fig. 8. Map of the sensitive bits obtained with the simulator for the 72Mult design. A close-up of the boxed region is shown in Figure 9

III. Radiation Testing

The purpose of this radiation test is to validate the results obtained from the simulator with the measurements obtained from the dynamic testing at the accelerator. Dynamic testing is performed in both the simulator and accelerator to illustrate that the functional sensitivity of a design is much less than what the static sensitive cross-section would suggest. In addition, the correlation of the accelerator test results with predictions from the simulator experiment demonstrates the accuracy of our technique. To perform dynamic testing, our objective was to operate the designs at speed (2 - 20 MHz) in the beam with the flux adjusted to slowly introduce upsets. The functional outputs as well as the configuration bitstream are continually monitored in the accelerator. When an error in the configuration bitstream or functional output is identified, the error is recorded and bitstream errors are repaired. The upset rate was adjusted to approximately one half the observation rate to limit the accumulation of upsets between observation cycles. This more closely represents the environment a design experiences on-orbit, where we assume that no more than one upset occurs in the bitstream before it can be repaired. Protons were chosen as a radiation source because they can provide a much lower effective rate of upset introduction, due to their lower interaction rate, than a heavy ion accelerator. Earlier work[2] demonstrated that the device is sensitive to protons, so no portion of the circuits was untested due to insufficient linear energy transfer (LET).

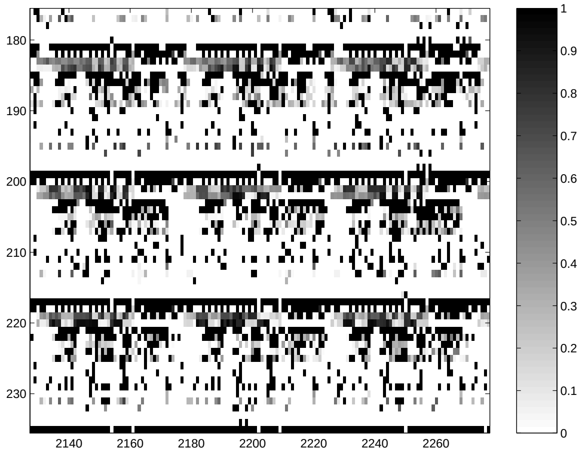


Fig. 9. A close-up of the map of sensitive bits within the 72Mult test design, as indicated in Figure 8. The probability of a design failure due to the upset of a given bit is illustrated by the shade of that bit. The likelihood of failure appears to be dependent not only upon site utilization, but also upon device architecture.

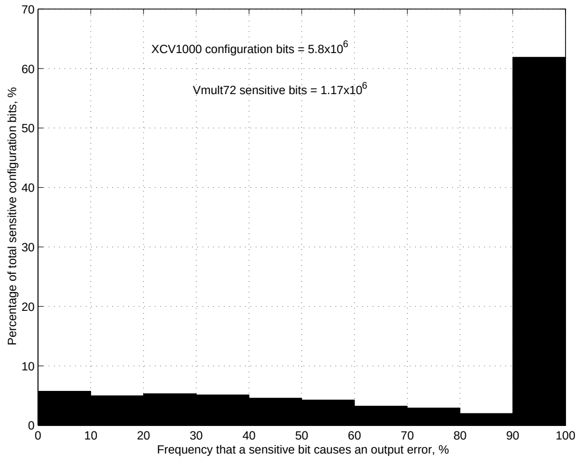


Fig. 10. Histogram of probability of failure of sensitive bits within the 72Mult test design ($1.17x10^6$ total sensitive bits). These results are obtained with the SEU simulator; they illustrate that the majority of sensitive bits cause a design failure 100% of the time.

The observation speed is limited by readback of the entire configuration bitstream of the Virtex 1000 on the SLAAC1-V. The observation cycle is about .5 seconds, so an average configuration upset rate of about one per second is desired. A $63.3MeV$ proton source was used and the beam flux was varied between $1.0 \times 10^7 \frac{p}{cm^2 \cdot s}$ and $3.5 \times 10^7 \frac{p}{cm^2 \cdot s}$. Even with an average upset rate of one configuration bitstream SEU per observation cycle, on many cycles multiple upsets were detected. There were many observation cycles with no observed upsets. Figure 11 shows the distribution of upsets per observation cycle.

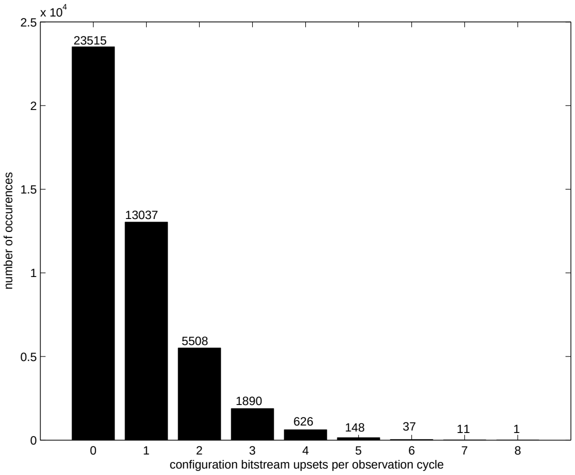


Fig. 11. Upsets per observation cycle for the 72Mult test design. A total of 44,773 observation cycles are included in this data.

The actual test setup used at the proton accelerator is shown in Figure 12. The SLAAC1-V PCI card was placed in a Linux PC by means of a PCI extender card. PE1, the socketed DUT, is irradiated with protons. A socket was used for the DUT so that the Virtex part could be exchanged if necessary, rather than trading out an entire board. However, only a single part was required: an XCV1000FG680 AFP0017 F1102747A 5C. The remainder of the test equipment, the PC and other portions of the SLAAC1-V, were protected from the beam by the means of a $.75''$ aluminum shield.

During the radiation testing with the SLAAC1-V board, the output of the DUT is compared with the output of the Golden Design. If there is ever a difference between the two, an output error is recorded along with the time stamp of the occurrence. In a similar fashion, the configuration bitstream is read back and monitored at a regular interval. If an upset is ever detected, its location is recorded with a corresponding time stamp. Subsequently, a partial reconfiguration of the device is used to repair the state of the configuration bitstream. If an output error was observed, both designs are then issued a reset in order to resynchronize their behavior. This process is outlined in Figure 13, and requires about $430ms$, on average, to complete.

Output errors observed at the accelerator are grouped into two categories: "predicted" and "flip-flop". "Predicted" output errors are those that originate from a bitstream upset identified by the simulator as sensitive. All other output errors from the accelerator are classified as "flip-flop" errors. An example of accelerator test data is shown in Figure 14.

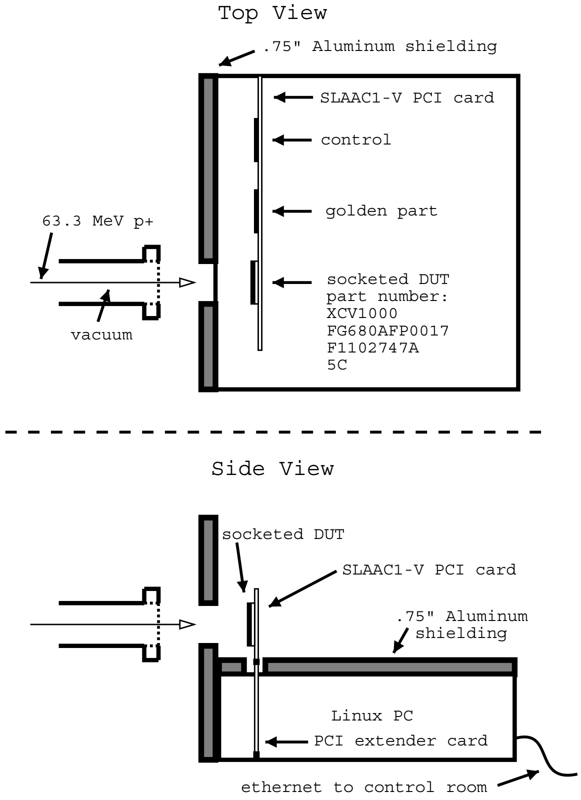


Fig. 12. Accelerator Test Setup.

Figure 14 also illustrates the decision process for determining the cause of an output error. The first highlighted example shows that the accelerator software detected an output error and a configuration bitstream upset on the same time stamp. With a $430ms$ granularity for time stamp measurements, this does not guarantee that the configuration bitstream upset was the cause of the output error. However, upon reference to simulator data, the configuration bitstream upset in question is shown to have caused an error 100% of the time during SEU simulations. If a configuration bitstream upset in question ever causes an error during an SEU simulation, it is assumed that it is responsible for any corresponding output error and classified as "predicted".

The third highlighted example in Figure 14 shows how a configuration bitstream upset and an output error occurring at the same time stamp do not always guarantee that an output error occurred due to a configuration bitstream upset. In this case, the output error was observed on the same time stamp as two other configuration bitstream upsets. However, because neither of these bitstream locations ever caused an error in any of the SEU simulations, the output error is classified as a "flip flop" error.

The final highlighted example in Figure 14 illustrates that a configuration bitstream upset and an output error do not always have to be observed on the same time stamp in order to be related. The output error was observed on the cycle immediately after the configuration bitstream upset, in this case a time difference of $426ms$. The simulator indicates

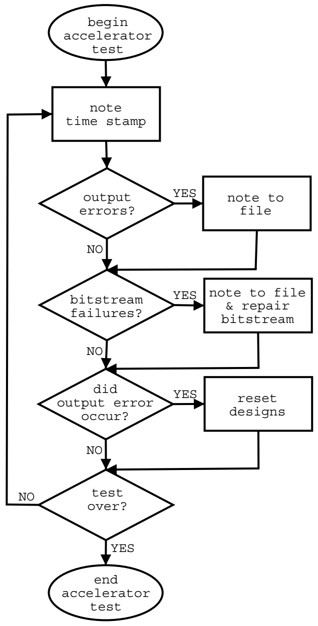
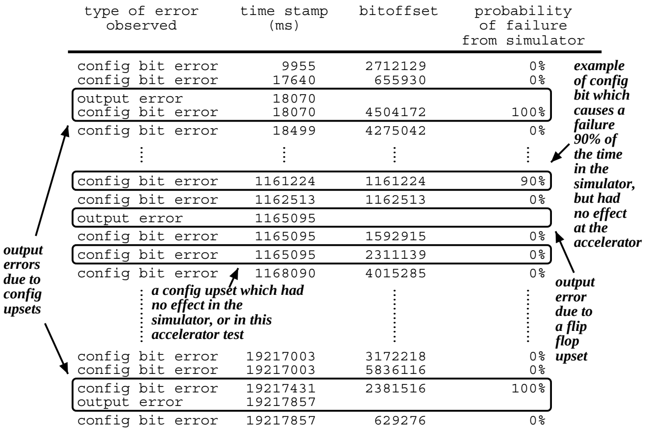


Fig. 13. Accelerator Testing Flow Diagram, each observation cycle ≈ 430 ms.



type of error observed	time stamp (ms)	bitoffset	probability of failure from simulator
config bit error	9955	2712129	0%
config bit error	17640	655930	0%
output error	18070		
config bit error	18070	4504172	100%
config bit error	18499	4275042	0%
⋮	⋮	⋮	⋮
config bit error	1161224	1161224	90%
config bit error	1162513	1162513	0%
output error	1165095		
config bit error	1165095	1592915	0%
config bit error	1165095	2311139	0%
config bit error	1168090	4015285	0%
⋮		⋮	⋮
config bit error	19217003	3172218	0%
config bit error	19217003	5836116	0%
config bit error	19217431	2381516	100%
output error	19217857		
config bit error	19217857	629276	0%

example of config bit which causes a failure 90% of the time in the simulator, but had no effect at the accelerator
output errors due to config upsets
a config upset which had no effect in the simulator, or in this accelerator test
output error due to a flip flop upset

Fig. 14. Sample accelerator test data with simulator information. This illustrates the classification of output errors as ‘predicted’ or ‘flip-flop’.

that the configuration bitstream location in question caused an output error 100% of the time during SEU simulations. A configuration bitstream upset can be observed before its effects are propagated to cause an output error. For this reason, whenever the upset of a bit known to cause an output error during SEU simulation is observed on the cycle immediately preceding that of an output error, we assume that this is a “predicted” output error.

Test coverage is an important consideration for dynamic testing. On-orbit, and in the accelerator, configuration upsets persist only briefly (.5 seconds or less) before being repaired. This implies that the upset is only tested against a tiny fraction of possible input vectors. We used LFSRs to generate our input test patterns for the multiplier test designs so that a large test space could be explored. The 72Mult test design has 2^{36} possible input vectors. In .5 seconds that translates to 10^7 input vectors tested against any particular upset when presenting inputs at 20MHz. This represents a coverage of .015%, which will only expose the most sensitive bits in the design. Increasing coverage at the accelerator is not practical because of the test time required.

The simulator has the flexibility to allow upsets to persist any amount of time longer than 215 μs. At the fastest loop time, the coverage is $6x10^{-6}$%. It is possible to extend coverage in the simulator. Just under an hour is required for the simulator to cover the entire input vector space for one configuration SEU, so covering more inputs for a few specific configuration bits is possible. Covering the entire input space for all the configuration bits is impractical.

A different perspective of coverage should be considered, and that is the number of configuration bits tested. Testing two configuration bits per second in the accelerator requires 800 hours of beam time to cover the entire configuration; that assumes each bit is upset once, which will not be the case in a random experiment. The simulator can offer an advantage in that every configuration bit can be tested. At the fastest possible rate, this requires about 20 minutes.

IV. Results

Our results for the accelerator experiment and corresponding simulator data are summarized in Table II. Each row of the table displays results for one of the three test designs, and was expected to vary with device utilization and logic style. Column 1 indicates the test design. Column 2 shows the number of configuration bitstream upsets detected for the test design while in the cyclotron. Columns 3 and 4 show the total number of output errors for the test design classified into those predicted by the simulator and flip-flop errors. Recall from the earlier discussion that the simulator does not cover flip-flop errors, so there is no corresponding column in the simulation results. Column 5 gives the percentage of configuration bitstream upsets that resulted in an output error, the ratio of columns 2 and 3. Column 6 is the average fluence to output error for each test design. Column 7 is the average fluence to configuration bitstream error. This is expected to be a constant measurement across a particular lot of silicon die from the same process and mask set. This measurement would be done once, then incorporated into the simulation results to arrive at the predicted fluence to output error in column 11. We will discuss this more later. Column 8 shows the number of configuration upset trials in the simulator, which are uniformly distributed throughout the bitstream. Column 9 shows the number of output errors generated from the upset trials in column 8. Column 10 is the percentage of the bitstream determined by the simulator that results in output errors, this is simply the ratio of columns 8 and 9.

There are two approaches to compare the simulator to the cyclotron. One simple comparison is to examine the number of predicted output errors to flip-flop errors for the 72Mult test design. Approximately 97% of the output errors were predicted by the simulator. The number of flip-flop errors for the other two test designs is insufficient to draw the same conclusion. We can, however, observe that the percentage of

Accelerator							Simulator			
1	2	3	4	5	6	7	8	9	10	11
Design	CBUs	POEs	Flip-Flop Errors	% of sensitive CBs	Avg Fluence to OE	Avg Fluence to CBU	CBUs	OEs	% of sensitive CBs	Pred Avg Fluence to OE
72Mult	33277	4958	108	14.9%	$7.8x10^7$	$1.2x10^7$	$2.32x10^9$	$3.6x10^8$	15.4%	$7.5x10^7$
36Mult	3003	146	2	4.9%	$2.6x10^8$	$1.3x10^7$	$2.32x10^9$	$1.0x10^8$	4.3%	$2.7x10^8$
72LFSR	1069	51	2	4.8%	$2.0x10^8$	$9.8x10^6$	$1.74x10^9$	$9.2x10^7$	5.3%	$2.2x10^8$

CBU - Configuration Bitstream Upset, POE - Predicted Output Error, OE - Output Error, CB - Configuration Bit

TABLE II

PROTON DYNAMIC TESTING RESULTS VS. SIMULATOR RESULTS FOR 3 TEST DESIGNS, FLUENCE IN P^+/cm^2.

sensitive configuration bits for each test design is extremely close (within 1%) to the percentage measured by the simulator.

Another comparison, cyclotron to simulator, is to examine the distribution of the fluence to output error. The cyclotron results are immediately available from the experimental data. The distribution of cyclotron fluence to output error for the 72Mult test design is plotted at the top of Figure 15. The average of the cyclotron obtained distributions for each test design are shown in column 6 of Table II. We can compare these to the simulator results. To obtain the same information from the simulator we used a sequence of upset trials and assigned each trial the weight of $1.17x10^7 P^+/cm^2$ of fluence. This figure is the weighted average in Table II column 7. This fluence to configuration upset measurement would only need to be made once on the mask set for a given process, and could be used subsequently in calculations for arbitrary dynamic designs. The same approach would be used for heavy ions. Using this approach, we transition the simulator data into fluence to output experiments. The fluence to output error distribution for the 72Mult test design obtained with the simulator is shown at the bottom of Figure 15. The simulator distribution averages are shown in column 11 of Table II and can be compared with cyclotron values in column 6. The averages shown here are within 10% of one another, suggesting the simulator is accurate to the first order at least.

Also observe the relative sensitivities among the three test designs. While more designs are required to quantify the trends, one can infer that sensitivity appears correlated to design utilization. The difference in the results for the LFSR and Mult designs suggests that different types of logic may result in different sensitivity.

V. CONCLUSION

The accelerator experiment measured the fluence to output error of a dynamic design under test, while recording the location and time of configuration bitstream upsets that occurred. The simulator generates a database of probabilities that a configuration bitstream upset within the DUT will result in a output error. Output errors observed at the accelerator were classified as “predicted” if they occurred near (in time) configuration upsets marked as “sensitive” by the simulator. Remaining output errors are classified as “flip-flop” errors. The data collected from the cyclotron experiment and the simulator runs substantially overlap. The simulator predicted 97% of the output errors observed in the accelerator experiment for the 72Mult test design.

There are advantages to using a simulator over accelerator experiments to test dynamic designs. One advantage is cost. While the development of the simulator involved extensive code development and acquisition of the simulation hardware, a similar amount of effort was needed to prepare for the accelerator experiment. This accelerator experiment generated approximately 40000 data points total for three test designs. The simulator generated millions of data points at no incremental cost for a variety of designs. A cost savings is realized as the simulator is used on more designs or to increase coverage of existing designs.

The motivation for the simulator development was to have a platform on which we could affordably explore various SEU mitigation concepts. While the FPGA manufacturer recommends TMR to harden designs against SEUs, there may be other techniques that trade power and bandwidth for reliability. We anticipate that many applications of FPGAs in space will need to trade reliability for performance. The simulator gives us the ability to quickly experiment with different points in the reliability spectrum. In addition, it can provide assurance that the mitigation the designer believes he has incorporated into the design actually makes it through the synthesis process. Another advantage of the simulator is that specific bits can be targeted and orchestrated with presentation of test vectors to achieve desired coverage. The simulator can guarantee that every configuration bit is tested. This is a powerful feature when assuring that a particular design is hardened against SEUs. These advantages do not eliminate the need for accelerator experiments, but the simulator has an important role in the experimental strategy.

The simulator generates a map of sensitive bits that can be used on-orbit, in conjunction with readback, to decide whether a design reset is required. The simulator output can also be used, in conjunction with an environmental radiation model and the static bit cross-section data, to more accurately predict the *in situ* dynamic design failure rate. The static cross-section only needs to be measured once on a process or at most on a per lot basis, not for every design intended for the devices.

The simulator is accurate to the first order at estimating the contribution of the bitstream to the dynamic sensitive cross-section of a design. We have shown that the configuration bitstream dominates the cross-section of the Virtex FPGA and

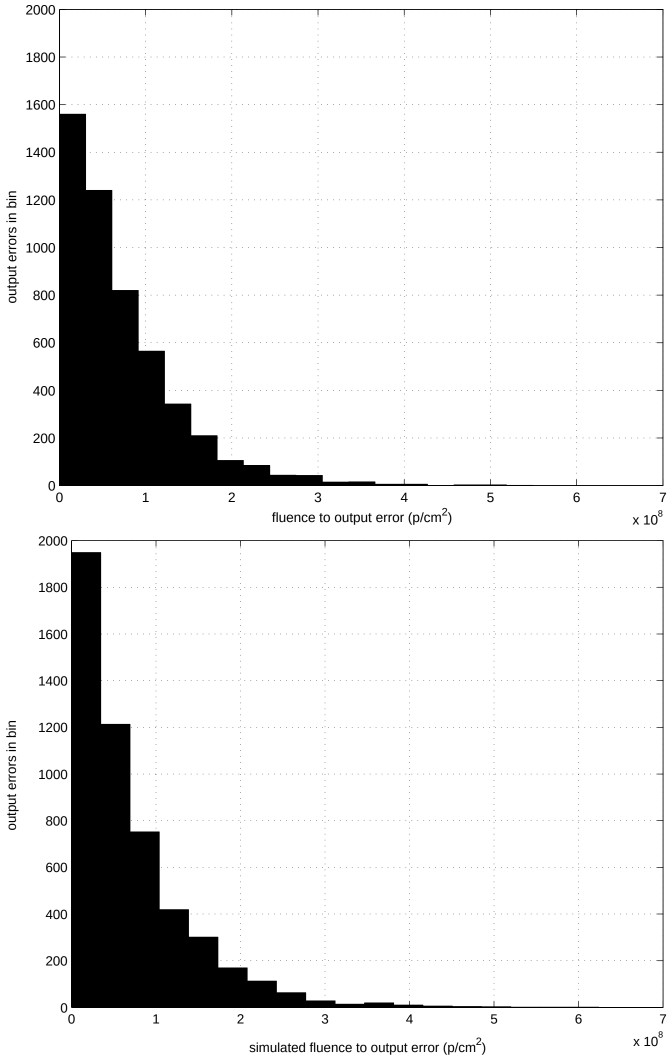


Fig. 15. Histogram of fluence to output error for 72Mult design. The top histogram shows data obtained from accelerator testing. The average fluence to output error at the accelerator for this design was $7.8 \times 10^7 \frac{p}{cm^2}$. The bottom histogram was obtained from simulator data, and assumes a fluence to configuration bitstream upset of $1.2 \times 10^7 \frac{p}{cm^2}$, as indicated in Table II. Using this data, the simulator reports and average fluence to output error of $7.5 \times 10^7 \frac{p}{cm^2}$. This illustrates the simulator's ability to forecast a dynamic sensitivity.

that the dynamic cross-section may be substantially smaller than the static cross-section.

VI. Acknowledgment

The authors would like to thank Xilinx, specifically, Carl Carmichael and Joe Fabula for their support of this work. We would also like to thank the University of Southern California's Information Sciences Institute (USC/ISI) for their support of our SEU simulation work. Finally, we like to acknowledge the Department of Energy's funding of this work through the Deployable Adaptive Processing Systems and Cibola Flight Experiment projects.

References

[1] M. Caffrey, "A space-based reconfigurable radio," in *Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA)*, T. P. Plaks and P. M. Athanas, Eds. CSREA Press, June 2002, pp. 49–53.

[2] E. Fuller, M. Caffrey, A. Salazar, C. Carmichael, and J. Fabula, "Radiation testing update, SEU mitigation, and availability analysis of the Virtex FPGA for space reconfigurable computing," in *MAPLD Proceedings*, September 2000.

[3] P. Graham, M. Caffrey, D. E. Johnson, N. Rollins, and M. Wirthlin, "SEU mitigation for half-latches in Xilinx Virtex FPGAs," in *IEEE Transactions on Nuclear Science*, vol. 50, no. 6, December 2003, submitted.

[4] P. Graham, M. Caffrey, M. Wirthlin, E. Johnson, and N. Rollins, "Reconfigurable computing in space: From current technology to reconfigurable systems-on-a-chip," in *24th Annual IEEE Aerospace Conference*, 2003, to be published.

[5] E. Johnson, M. J. Wirthlin, and M. Caffrey, "Single-event upset simulation on an FPGA," in *Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA)*, T. P. Plaks and P. M. Athanas, Eds. CSREA Press, June 2002, pp. 68–73.

[6] M. Wirthlin, E. Johnson, N. Rollins, M. Caffrey, and P. Graham, "The reliability of FPGA circuit designs in the presence of radiation induced configuration upsets," in *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines (FCCM '03)*. IEEE Computer Society, 2003, to be published.

[7] *SLAAC-1V User VHDL Guide*, USC-ISI East, October 1, 2000, release 0.3.1.

[8] M. Geaorge and P. Alfke, "Linear feedback shift registers in Virtex devices," Xilinx Corporation, Tech. Rep., January 9, 2001, xAPP210 (v1.0).

[9] C. Carmichael, M. Caffrey, and A. Salazar, "Correcting single-event upsets through Virtex partial configuration," Xilinx Corporation, Tech. Rep., June 1, 2000, xAPP216 (v1.0).

Eric Johnson is a graduate student in the electrical and computer engineering program at Brigham Young University with interests in digital signal processing and reconfigurable computing.

Michael Caffrey is a member of the technical staff at Los Alamos National Laboratory. His current work involves the production of a FPGA-based reconfigurable computing platform for space and has interests in reconfigurable computing and digital signal processing. Mr. Caffrey obtained an M.S. degree in electrical engineering from the University of New Mexico in 1995.

Paul Graham is a member of the technical staff at Los Alamos National Laboratory. His research interests include design automation tools for FPGA-based reconfigurable computing, applications of reconfigurable computing, and FPGA architecture. Dr. Graham obtained an M.S. degree in electrical engineering from Brigham Young University in 1996, and a Ph.D. in Electrical Engineering from Brigham Young University in 2001.

Nathan Rollins is an undergraduate student in the electrical and computer engineering program at Brigham Young University with interests in reconfigurable computing.

Michael Wirthlin is an assistant Professor at Brigham Young University in Provo, UT. His research interests include tools and applications for FPGA-based reconfigurable computing, power efficiency in FPGA designs, and synthesis techniques for digital systems. Dr. Wirthlin obtained a Ph.D. in electrical engineering from Brigham Young University in 1997.